IN THE CLAIMS

1-21. (cancelled)

22. (previously presented) A method comprising:

generating, by a first vulnerability analysis and fortification (VAF) agent operating in a hardware-based system, a first process representation of a first process, wherein the first process comprises a series of sequential operations that are represented by multiple nodes in the first process representation, and wherein the first VAF agent monitors the first process for security exposures;

defining legal and illegal interfaces between the multiple nodes in the first process representation, wherein a legal interface between a first node and a second node in the first process representation reflects an authorization for operations represented by the first node and the second node to be linked, and wherein an illegal interface reflects a lack of authorization for operations represented by the first node and the second node to ever be directly linked;

generating, by a second VAF agent, a second process representation of a second process;

comparing nodes from the first process representation to nodes of the second process representation, wherein the second VAF agent monitors the second process for security exposures; and

in response to the nodes of the first process representation matching nodes in the second process representation, sending an alert from the first VAF agent to the second VAF agent, wherein the alert identifies the illegal interfaces between nodes in the first process representation as potential illegal interfaces between nodes in the second process representation.

23. (previously presented) The method of claim 22, wherein the legal interface further reflects a requisite action in the first node that is required to reach the second node, and wherein the illegal interface further reflects an absence of the requisite action in the first node.

24. (previously presented) The method of claim 23, wherein the legal interface further reflects a requisite return code being transmitted from the second node to the first node, wherein

the requisite return code is transmitted in response to a password being sent from the first node to the second node.

25. (previously presented) The method of claim 22, further comprising:

in response to the first VAF agent detecting the illegal interface, creating, by the first VAF agent, a security patch that prohibits the first node from being linked to the second node; and

transmitting the security patch from the first VAF agent to the second VAF agent.

26. (previously presented) The method of claim 22, wherein the first process and the second process are component parts of a single distributed software system.

27. (previously presented) The method of claim 22, wherein the first process representation and the second process representation are respectively derived from an extensible markup language (XML) description of the first process and the second process.

28. (previously presented) The method of claim 27, wherein the XML description of the first process is stored in a security server that is protected by a first firewall detection system, and wherein the XML description of the second process is stored in the security server and is protected by a different second firewall detection system.

29. (previously presented) The method of claim 22, wherein the method described in claim 22 is embodied as an add-on software component, the method further comprising:

adding the add-on software component to an enterprise solution before shipping the enterprise software solution to a customer, wherein the enterprise solution is based on the first process.

30. (previously presented) The method of claim 22, wherein the first VAF agent and the second VAF agent are controlled by a single VAF tool.

31. (previously presented) The method of claim 22, wherein the first process representation and the second process representation are depicted as graphs.

32. (previously presented) A computer program product comprising a computer readable storage medium embodied therewith, the computer readable storage medium comprising:

computer readable program code configured to generate, by a first vulnerability analysis and fortification (VAF) agent operating in a hardware-based system, a first process representation of a first process, wherein the first process comprises a series of operations that are represented by multiple nodes in the first process representation, and wherein the first VAF agent monitors the first process for security exposures;

computer readable program code configured to define legal and illegal interfaces between the multiple nodes in the first process representation, wherein a legal interface between a first node and a second node in the first process representation reflects an authorization for operations represented by the first node and the second node to be linked, and wherein an illegal interface reflects a lack of authorization for operations represented by the first node and the second node to be directly linked;

computer readable program code configured to generate, by a second VAF agent, a second process representation of a second process;

computer readable program code configured to compare nodes from the first process representation to nodes of the second process representation, wherein the second VAF agent monitors the second process for security exposures; and

computer readable program code configured to, in response to the nodes of the first process representation matching nodes in the second process representation, send an alert from the first VAF agent to the second VAF agent, wherein the alert identifies the illegal interfaces between nodes in the first process representation as potential illegal interfaces between nodes in the second process representation.

33. (previously presented) The computer program product of claim 32, wherein the legal interface further reflects a requisite action in the first node that is required to reach the second node, and wherein the illegal interface further reflects an absence of the requisite action in the first node.

34. (previously presented) The computer program product of claim 33, wherein the legal interface further reflects a requisite return code being transmitted from the second node to the first node, wherein the requisite return code is transmitted in response to a password being sent from the first node to the second node.

[[36]]35. (currently amended)The computer program product of claim 32, wherein the first process and the second process are component parts of a single distributed software system.

[[37]]36. (currently amended)The computer program product of claim 32, wherein the computer program product is an add-on software component, the computer program product further comprising:

computer readable program code configured to add the add-on software component to an enterprise solution before shipping the enterprise software solution to a customer.

[[38]]37. (currently amended)The computer program product of claim 32, wherein the first VAF agent and the second VAF agent are controlled by a single VAF tool.

[[39]]38. (currently amended)A system comprising:

a central processing unit (CPU), a computer readable memory, and a computer readable storage media;

first program instructions for generating, by a first vulnerability analysis and fortification (VAF) agent operating in a hardware-based system, a first process representation of a first process, wherein the first process comprises a series of sequential operations that are represented by multiple nodes in the first process representation, and wherein the first VAF agent monitors the first process for security exposures;

second program instructions for defining legal and illegal interfaces between the multiple nodes in the first process representation, wherein a legal interface between a first node and a second node in the first process representation reflects an authorization for operations represented by the first node and the second node to always be linked, and wherein an illegal interface reflects a lack of authorization for operations represented by the first node and the second node to ever be directly linked;

third program instructions for generating, by a second VAF agent, a second process representation of a second process;

fourth program instructions for comparing nodes from the first process representation to nodes of the second process representation, wherein the second VAF agent monitors the second process for security exposures; and

fifth program instructions for, in response to the nodes of the first process representation matching nodes in the second process representation, sending an alert from the first VAF agent to the second VAF agent, wherein the alert identifies the illegal interfaces between nodes in the first process representation as potential illegal interfaces between nodes in the second process representation, and wherein

said first, second, third, fourth and fifth program instructions are stored on said computer readable storage media for execution by said CPU via said computer readable memory.

[[40]]39. (currently amended)The system of claim [[39]]38, wherein the first process and the second process are component parts of a single distributed software system.

[[41]]40. (currently amended)The system of claim [[39]]38, wherein the first VAF agent and the second VAF agent are controlled by a single VAF tool.